NOTE XN: XNUMBERS - MULTI PRECISION FLOATING POINT CALCULUS FOR EXCEL

Version of July 20, 2006

THE VOLPI XNUMBER ADD-INS

(This is summary material extracted from the site http://digilander.libero.rt/foxes. It is presented as an explanation on how xnumbers works in an Excel environment.)

Benvenuti in questo sito che è stato costruito sotto la pressante richiesta dei nostri (tre o quattro) affezionati utenti. Soprattutto è stato fatto per permettergli di prelevare direttamente i nostri programmi senza vederli frugare nel nostro PC. A distanza di più di un anno dalla comparsa del primo pacchetto di aritmetica in virgola mobile a precisione variabile per Excel - **XNUMBERS** - divertiti (più che altro sorpresi) dal discreto successo di diffusione, ne abbiamo curato la nuova versione che è ora quasi presentabile. Cosa sia questo programma e a cosa serva è spiegato subito dopo.

Questo è anche il sito dove potete trovare il pacchetto per calcolo matriciale e algebra lineare per Excel **MATRIX** il cui successo ci ha divertito ancora di più. Tutte e due sono freeware.

Buon calcolo a tutti.

THE CURRENT ADD-INS

These are *.zip files that when downloaded and unzipped, generate *.XLA files and several useful files including tutorials and help files. They are accessible from http://digilander.libero.it/foxes/SoftwareDownload.htm. There are tutiorials also available.

Xnumbers is Freeware. and is Open Source. It is not restricted as to use (commercial, educational, experimental, etc.). If included in purchased, commercial Excel add-ins, the developer should obtain clearance from Professor Volpi.. **THIS IS THE ONLY KNOWN ADD-IN THAT ALLOWS UNRESTRICTED (NO PASSWORD BLOCKING) ACCESS TO THE VBA FUNCTIONS. This means that errors are freely found by users and corrected through new issues.**

XNUMBERSDLL

This is a set of multi-precision basic arithmetic functions in a series of *.dll files. These functions can only be accessed through other add-ins or by user written VBA functions. De Levie's add-in set (described below) requires the XNUMBERSDLL.ZIP download.

XNUMB46

XNUMBERS.XLA is an add-in that provides a large collection of Excel useable functions, including the multi-precision set of functions. The multi-precision functions can be accessed either as user written VBA functions or as direct worksheet cell equations. It consists of a set of more than 200 functions for **arithmetic**, **complex**, **trigonometric**, **logarithm**, and **exponential** calculus.

Linear Algebra: System solutions with a Gauss-Jordan algorithm (also in a didactic step-by-step method), and a Gauss-Seidel iterative method. Matrix operations. Determinant. Matrix Inversion. Characteristic Polynomial with Newton-Girard formulas, Crout's algorithm for LU decomposition. Similarity Transformation. Matrix power.

Polynomials: symbolic computation to add, multiply and divide polynomials. Derivatives of polynomials. Interpolation with Newton formula. Rootfinder algorithms: Newton-Raphson, Lin-Bairstow, Halley methods, also for didactic scope.

Numbers Theory: MCD, MCM, factoring, prime number, fraction reduction. **Statistic**: Mean, Variance, Multivariable Linear Regression, LRE.

Integration : Romberg , Newton-Cotes, Filon, FFT, DFT,

Numerical Series evaluation

Special functions: high precision 64b Bessel, Gamma, Gammaln, Digamma, Fresnel, integral sinus-cosinus, exponential integral, error function, Beta function, Fibonacci.

Interpolation : polynomial, fractional, cubic spline

MATRIX

Matrix 2.0. Matrix and linear algebra functions for Excel.

BIGMATRIS12

Bigmatrix 12. A large matrix elaboration for Windows OS and Excel.

OPTIMIZ_TOOL

An optimization function (marq and Levenberg type) with applications for Excel.

ORTHOPOLY

Deals with orthogonal polynomials and solutions for Excel.

FDSOLVER

Solves differential equations numerically by means of finite differences for Excel.

OTHER DOWNLOADS

There are others such as file conversions, math parsers, etc.

MULTI-PRECISION COMPUTING.

Xnumbers is an Excel add-in (xla) that contains functions that perform multi-precision floating-point arithmetic. The xnumbers add-ins also include other functions that do not use the multi-function routines. Perhaps it is the first package that extends the standard Excel precision from 15 up to 200 significant digits. It is compatible with Excel 97/98/2000/2003.

THEORY

Several methods exist for simulating variable multi-precision floating-point computations on 32-bit machines. The basic concept consists of breaking down long number as two or more sub-numbers and repeating cyclic operation with them. The way in which long numbers are stored varies from one method to another. Two of the most popular methods use the "string" conversion and "packing"

HOW TO STORE LONG NUMBERS

STRING EXTENDED NUMBERS

In this method, long numbers are stored as vectors of characters; each of them represents a digit to the base 256. During input, numbers are converted from decimal to 256 base and vice versa in output. All internal computation in 256 base requires only 16 bits for storing and a 32 bit accumulator for computing. An example of how to convert the number 456789 to a string follows:

$(456789)_{10} = (6,248,85)_{256}$

String= chr(6)&chr(248)&chr(85)

This method is very fast. Also efficient algorithms for the input-output conversion have been realized. A good explanation about this method can be found in "NUMERICAL RECIPES in C - The Art of Scientific Computing", Cambridge University Press, 1992, PP 920-928. In this excellent work you can find also efficient routines and functions to implement an arbitrary precision arithmetic. Perhaps, the only critical factor in this method regards the debugging and testing activity. It will be true that the computer does not take care of the base representation of numbers, but the programmer, usually, does it. During debugging, the programmer examines lots and lots of intermediate results and he must always translate them from base 256 to 10. It must be pointed out that, for this kind of programs, the debugging and tuning activity reaches normally 80 - 90% of the total development time.

PACKET EXTENDED NUMBERS

This method avoids the conversion the base representation of long numbers and stores them as vectors of integers.

This is adopted into all FORTRAN77 routines of "MPFUN: A MULTIPLE PRECISION FLOATING POINT COMPUTATION PACKAGE" by NASA Ames Research Center. For further details we refer to the refined work of David H. Bailey published in "TRANSACTIONS ON MATHEMATICAL SOFTWARE", VOL. 19, NO. 3, SEPTEMBER 1993, PP. 286-317.

Of course this add-in does not have the performance of the mainframe package (16 million digits) but the method is substantially the same. Long numbers are divided into packets of n° digits.

For example, the number 601105112456789 in packet form of 6 digits, become the follow integer vector:

456789
105112
601

As we can see, the sub-packed numbers are in decimal base and the original long number is perfectly recognisable. This a great advantage for the future debugging operation of algorithms.

An example of arithmetic operation - a multiplication A x B = C - between two packet number is shown in the following:

A		B
456789		654321
105112	X	
601		

The schema below illustrated the algorithm adopted:

carry		A		B		C'		C
0	+	456789	x	654321	=	298886635269	=>	635269
298886	+	105112	x	654321	=	68777287838	=>	287838
68777	+	601	x	654321	=	393315698	=>	315698
393	+	0	x	654321	=	393	=>	393

As we can see the maximum digit is reached into accumulator C'; in the other vectors, numbers require only six digits at max. In 32-bit systems, the maximum for single packets is 6 digits. This is equivalent to a conversion from decimal into a 10^6 representation base. The maximum number of digits for single packets depends of the hardware accumulator. Normally, for 32-bit computer architecture, the 6-digit form is adopted, but this value is not critic at all. Values from 4 to 7 affects the computation speed about 30 %. Values from 4 to 7 do not affect the precision of results in any case.

GENERAL DESCRIPTION

The package XNUMBERS.XLA has several functions covering the principal field of standard computation. The basic arithmetic functions: addition, multiplication, division are the most important. They accept two numbers and an optional parameter for setting the significant digits of computation. The functions recognise decimal format (12.34 , -0.056, etc.) and exponential format (1.234567E-015) as well. The functions accept also decimal, double, currency or any other Excel number; or even mixed (extended number + normal number).

There are functions for computing radix (squared, cubic, and n-th) and taking a number to the power of an integer; as well as the most important trigonometric and transcendental functions. Some useful constants like PI, Ln(2), Ln(10) with digits up to 400 are provided.

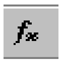
This package performs multi-precision floating-point arithmetic up to 200 significant digits. You can set a precision level separately for each function by an optional parameter. By default, all routines use the precision of 30 digits, but it can be easily changed continually from 1 to 200 significant digits.

The multi-precision functions do not use the Intel single and double precision machine instructions, doing everything in terms of characters operations. The result is a major reduction in computing speed. Therefore depending on the application, there will be a tradeoff between the extent of precision and the time imvolved in updating worksheets.

Computing time is of course increased when the number of digits is increased. Therefore the recommendation for only quadratic precision (30 decimals) has merit.

USE

These functions can be used in an Excel worksheet as any other built-in function. After the installation, look up in the functions library <Functions…> from the menu <Insert>, or click on the icon



Upon "user's" category you should find all the functions of this package.

From version 2.0 you can manage functions also by the Function Handbook starting by the following icon

All the functions for multi-precision computation begin with "x". The example below shows two basic functions for addition and subtraction operations.

	A	B
2	123456789,123456	
3	0,0123456789	
4	123456789,1358020000	=A2+A3
5	123456789,1358016789	=xadd(A2;A3)
6	123456789,1111100000	=A2-A3
7	123456789,1111103211	=xsub(A2;A3)
8		

As any other function they can also be nested to build complex expressions. In the example below we compute x^4 with a precision of 30 digit

	A	B
2	1234567	
3	2323050529221950000000000	=A2^4
4	2323050529221952581345121	=xmult(A2;xmult(A2;xmult(A2;A2)))
5		

You can also insert extended numbers directly in the function. Only remember that for preserving the character form in Excel, you must insert them like a string, within quotation marks (example. "134.3915574").

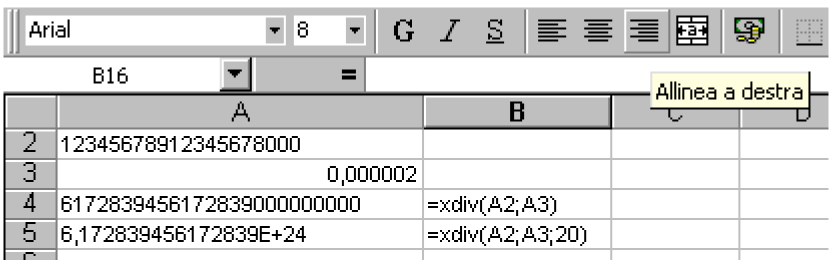
xpow("0.391557496360989810771470166011";90) =
1.99045089214781765089811552 84E-7

If you want to insert a long number in a cell and you do not want Excel to automatic convert it, than prefix it with the apostrophe symbol < ' >.

PRECISION

All the mult-precision functions in this package have an optional parameter - Digit_Max - setting the maximum of significant digits for floating point computation. In this version the maximum precision level has been set to 200 digits, and the default as 30. This parameter also determines how the output is automatically formatted. If the result has less digits than Digit_Max, then the output is in the plain decimal format (123.45, -0.0002364, 4000, etc.); else if the result exceeded the max digits allowed (significant or not), the output is automatically converted in exponential format (1.23456789E+94). The output format is independent of input format.

In the example below, the result, A4=A2/A3, needs 25 digits in plain decimal format (note that significant digit are only 16). In the cell A4 we can see just this format as we have set Digit_Max=30; on the contrary, in cell A5, for the same operation, we see the exponential format as we have set Digit_Max=20. Note here that the comma in the numbers is the European expression for the decimal point.



In synthesis, the Digit_Max parameter limits:

- The significant digit of internal floating point computation
- The max digits to output, significant or not.

FORMATTING THE RESULT

At this version the user cannot format an extended number with standard Excel number format tools, as the x-numbers are for Excel simply a string of characters. In fact, as we can see in the example above, Excel aligns the x-number, on the left, as normal string. To change the alignment, use the standard Excel format tool.

From version 2, it is possible to separate the digits of a string numbers in group, by the user function **xFormat()** and **xUnformat()** . They work similar at the built-in function Format(x, "#,##0.00"). These functions were original developed by Ton Jeursen for his add-in XNUMBER95, the downgrade version of XNUMBERS for Excel 5. Because they work well and they are very useful for examining long string of number, I have imported them also in this package.

MORE INFORMATION

Download "**Tutorial of Numerical Analysis with Xnumbers.xla** -Vol 1. Jan 2006, PDF (2.2 MB)

Il NOSTRO TEAM (THE FOXES)

Ebbene sì. Siamo al "Redde Rationem".

I redattori nonché gestori di questo sito sono:

Irene Menicagli, giovanissima studentessa di Matematica

Leonardo Volpi, né giovane, né studente, colpevole ideatore di tutto questo

Per contattarci inviate a: leovlp@libero.it ,

OUR FRIENDS…

I nostri amici e collaboratori.(in ordine di apparizione) che ringraziamo per la loro cortese collaborazione fatta di suggerimenti, idee, routines, software, links, incoraggiamenti, segnalazioni di bug, documenti, e molto altro.
A loro inviamo i nostri ringraziamenti e quelli di tutti quanti ci hanno scritto:

XNUMBER STATISTICAL FUNCTIONS

The add-ins are primarily for mathematical uses. However there are some functions that would be of interest for statistical calculations.

Table 1.1 Xnumber Statistical Functions

Name	Ref	Related Excel Function	XNUMBER Function	Mult-Precision
Maximum	1.1	MAX	xmax	No
Minimum	1.2	MIN	xmin	No
Arithmetic Mean	1.3	AVERAGE	xmean	Yes
Geometric Mean	1.4	GEOMEAN	xgmean	Yes
Quadratic Mean	1.5		xqmean	Yes
Variance	1.6	VAR	xvar	Yes
Standard Deviation	1.7	STDEV	xstdev	Yes
Factorials	2.1	FACT	xfact	Yes
Double step factorial	2.2	FACTDOUBLE	xfact2	Yes
Combinations	2.3	COMBIN	xcomb	Yes
Combinations of large Numbers	2.4		xcomb_big	No
Permutation	2.5	PERMUT	xperm	Yes
Beta Distribution	3.1	BETADIST	xbeta	No
Gamma Distribution	3.2	GAMMADIST	xGamma	No
Fisher's Gamma Distribution	3.3		xGammaF	No
Hypergeometric Distribution	3.4	HYPGEOMDIST	Hypergeom	No
Gauss Error Function	3.5	ERF	errfun	No
Log Relative Error	4.1		xLRE	No

Log Gamma (natural)	4.2	GAMMALN	xGammalog	No
Coefficients of Linear Regression	5.1	LINEST	RegLin_Coeff	No
Coefficients of Linear Regression	5.2		xRegLin_Coef	Yes
Linear Regression With Robust Method	5.3		RegLinRM	No
Linear Regression Min-Max	5.4		RegLinMM	No
Returns Linear Regression R^2 Value	5.5		RegLinStat	No
Returns Linear Regression R^2 Value	5.6		XRegLinStat	Yes
Evaluates Linear Regression at Point x	5.7		RegLin_Eval	No
Evaluates Linear Regression at Point x	5.8		XRegLin_Eval	Yes
Converts Coefficients of Conditioned Data to Coefficients of Row Data	5.9		RLCondCoef	No
Returns Linear Regression Covariance Matrix	6.1	Data Analysis Covariance	RegLin_Covar	No
Returns Linear Regression Covariance Matrix	6.2		xRegLin_Covar	Yes
Optimization, Downhill Simplex	7.1	Solver	Optim1	No
Optimization, Divide and Conquer	7.2	Solver	Optim3	No

OPTIMIZATION TOOL

OPTIMIZ.XLA - VER. 2.0 - MAY. 2006

New macros added:
Nonlinear system of equations solvers
Newton-Raphson = NL-system solver with Newton algorithm
Broyden = NL-system solver with Broyden algorithm (thanks to J.L. Martinez)
Brown = NL system solver with Brown algorithm
Global = Searches for all roots in a given range
1D-Zerofinder = Miscellanea of zerofinders for nonlinear univariate equations
2D-Zero path = Zero contour finder for bivariate equations f(x,y)=0
2D intersection = Finder of intersections between two zero contours
LP with Simplex algorithm
NLP with linear constraints
Logistic regression
Rational regression with separated degrees for numerator and denominator
Exponential regression up to 6 parameters
Multi variables regression LM (from 1 to 9 parameters)
LM regression now computes the Standard Deviation of the Estimates (issue by Albert Bemmaor)
The globally accuracy of the LM algorithm and CG algorithm was improved

Many thanks also to D. A. Heiser for his kind contribution in testing, debugging and documentation revision

LEVENBERG-MARQUARDT ALGORITHM

NONLINEAR REGRESSION

Regression analysis for non-linear functions is not an easy subject.

Technically speaking, Nonlinear Regression is a general fitting procedure that will estimate any kind of relationship between a dependent (or response variable), and a list of independent variables. In general, all regression monovariable models may be stated as:

$$y^* = F(x, c_1, \dots , c_n)$$

The goal of regression analysis is to determine the values of parameters: c_1 c_2... c_n, for a function that cause the function to best fit a set of data observations that you provide. The term "best fit" means to determine the values of the parameters that minimize the sum of the squared residual values for the set of observations. This is known as a "least squares" regression fit.

Levenberg-Marquardt is a popular alternative to the Gauss-Newton method of finding the minimum of a function $R(u_1, \dots u_n)$ that is a sum of squares of nonlinear functions. In our case, R is the sum of square residual error of the regression. That is

$$R(c_1, \dots , c_n) \;=\; \Sigma\,(y_i - y^*(x_i)) \;=\; \Sigma\,(y_i - F(x_i, c_1, \dots , c_n))$$

This algorithm is very efficient and fast having also a quite good global convergence property. However finding this algorithm (and also others related to minimization task) on public domain is not very easy.

Our team, thanks to *Luis Isaac Ramos Garcia*, can now publish a version of Levenberg-Marquardt algorithm for Visual Basic. (Levenber_Marquaurdt.bas). The original version, developed by *Luis,* is very compact. It needs only the module for solving linear system (Solve_Linear_Sistems.bas) also developed by *Luis*.

The actual version is updated 9.4.2006

The algorithm is performed by the

Sub **LMNoLinearFit**(x() As Double, y() As Double, c() As Double, ch2 As Double)

Where:

x() = vector of x-data to fit

y() = vector y-data to fit

c() = vector of parameters to optimize. At the begin contains the starting point

ch2 = at the end contains the square residual error

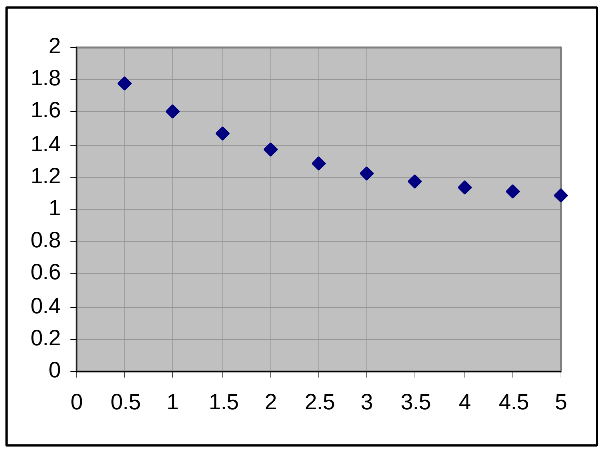
Before calling this sub you must provide the definition of the function and its derivatives

```
Function FUN(x, c)    'function to minimize
Sub DFun(x, c, Dv)    'derivatives of function to minimize
```

Now let's see with an example how it works

Assume to have the following data set

x	y
0.5	1.77880078307140
1	1.60653065971263
1.5	1.47236655274101
2	1.36787944117144
2.5	1.28650479686019
3	1.22313016014843
3.5	1.17377394345045
4	1.13533528323661
4.5	1.10539922456186
5	1.08208499862390



We want to find an exponential function of the following type that best fit the given points (x, y)

$$F(x, c_1, c_2) = \exp(c_1 * x) + c_2$$

The parameters c_1 and c_2 are unknowns for the moment. Because the algorithm needs a starting point for searching the best fit we can initialize them with:

$c_1 = 0$, $c_2 = 0$ (Tip: usually we should provide a sufficiently approximate value for the starting point)

Now we have to obtain the derivatives of the function $F(x, c_1, c_2)$ respect to each parameter

$$dF/dc_1 = x * \exp(c_1 * x)$$

$$dF/dc_2 = 1$$

A simple driver program for the **LMNoLinearFit** subroutine can be the following

```
 Function FUN(x, c)
'function to minimize
    FUN = Exp(c(1) * x) + c(2)
End Function

Sub DFun(x, c, Dv)
'derivatives of function to minimize
ReDim Dv(1 To UBound(c))
    Dv(1) = x * Exp(c(1) * x)
    Dv(2) = 1
End Sub
```

```
Sub NLfit_test1()
Dim tmp, x() As Double, y() As Double, c() As Double, N As Long, i As Long, ch2
As Double
'load y data
tmp = Array(1.7788007830714, 1.60653065971263, 1.47236655274101, _
            1.36787944117144, 1.28650479686019, 1.22313016014843, _
            1.17377394345045, 1.13533528323661, 1.10539922456186,
1.0820849986239)

N = UBound(tmp) + 1
ReDim x(1 To N), y(1 To N)
For i = 1 To N
    y(i) = tmp(i - 1)
Next i

'Load x data
tmp = Array(0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5)
For i = 1 To N
    x(i) = tmp(i - 1)
Next i
```

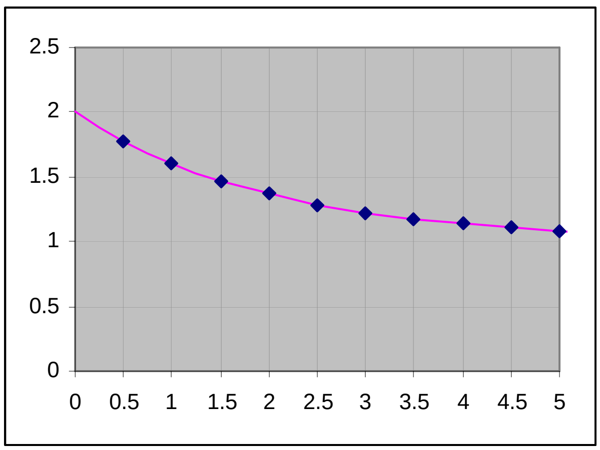
```
ReDim c(1 To 2)
'initialize with a starting point (c1, c2) that you like
c(1) = 0: c(2) = 0

'find the best fit for f(x, c1, c2) = exp(c1*x) + c2
Call LMNoLinearFit(x, y, c, ch2)

'Output results
Debug.Print "Non-linear regression of:  exp(c1*x) + c2"
Debug.Print "c1 = "; c(1), "c2 = "; c(2), "Error = "; Sqr(ch2)

End Sub
```

The ouput will be

```
Non-linear regression of:  exp(c1*x) + c2
c1 = -0.499999999999997     c2 =  0.999999999999997     Error =
9.29675066505912E-15
```

Plotting the function

F(x) = exp(-0.5*x)+1

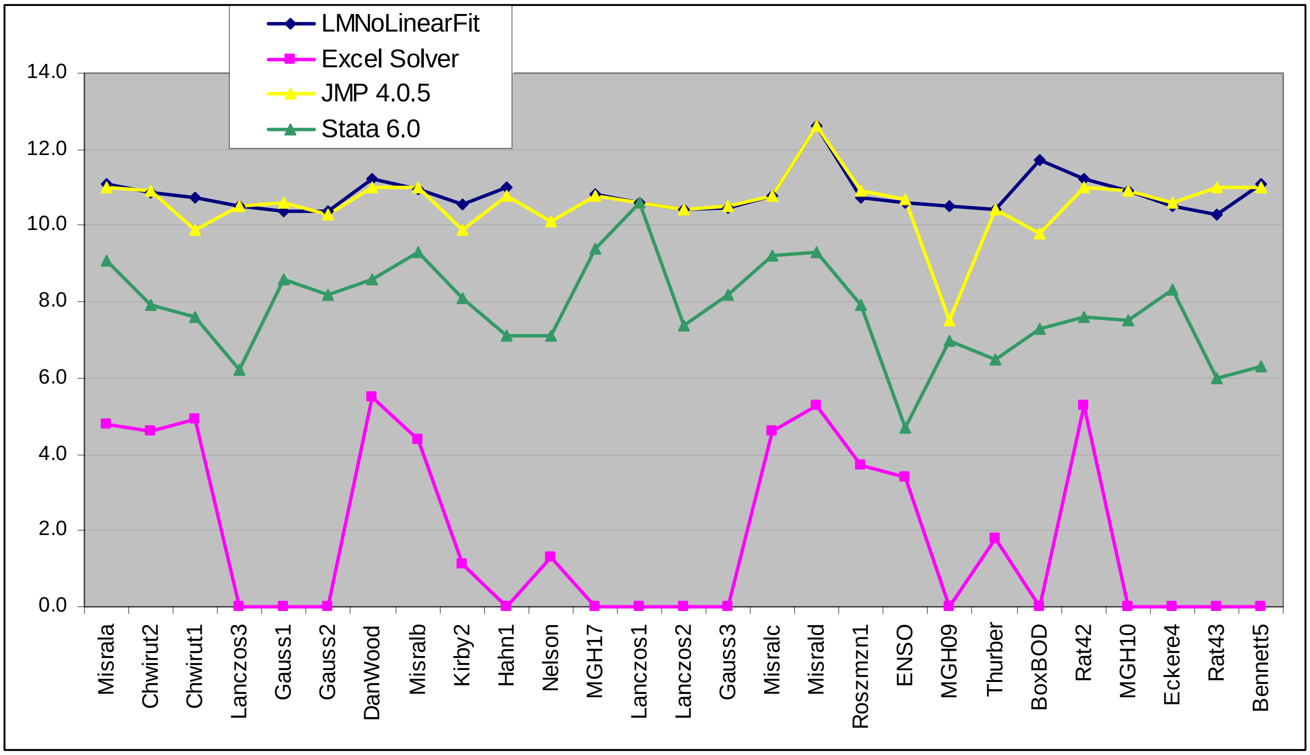


We note a very good fitting. Not always of course the model fits so perfectly to the given data. You have to do many trials before reaching a sufficiently accurate model.

But this smart software can help you in this research.

Recently, thanks to *David A. Heiser*, the core LM routine **LMNoLinearFit** was improved and tested with the non-linear NIST StRD dataset

The test result - the minimum LRE for all regressors - is reported in the following graph, compared with the the **Solver** of Excel 97, **JMP** 4.0.5 (Creighton and Ding, 2000), and **STATA** 6.0. (STATA web pages, 1999)



ROBERT DE LEVIE'S MACROS

Robert De Levie discovered the xnumbers.dll source, and programmed up a number of functions useful for statistical analysis. The can be found at his site, http://www.bowdoin.edu/~rdelevie/excellaneous.
They are all VBA functions that reference the xnumbers.dll download. Again, these VBA functions can be examined, since they are not password protected, and are freely available. However Robert De Levie has claimed copyright protection, and any commercial use of them should have his permission. They go beyond the Volpi sets in some ways, and actually a combination of the two would be most advantageous.

THE CONTENTS OF THE XMACROBUNDLE

There are two files of VBA functions that can be downloaded. The MacroBundle (http://www.bowdoin.edu/~rdelevie/excellaneous/macrobundle53.doc) and the xMacroBundle (http://www.bowdoin.edu/~rdelevie/excellaneous/macrobundle55.doc). The Macrobundle and xMacrobundle VBA sets should be read before being used. Those that are to be used are cut from the Word form and pasted into one of the VBA project directories. The narratives within the functions are all string identified.

The xMacroBundle is a work in progress, and will contain extended-precision versions of several macros in the MacroBundle compatible with the extended precision freeware from Xnumbers. Because these extended-precision macros are considerably slower than their regular, double-precision versions, they are *not* intended for routine applications.

After you download and install xnumbers.dll, you can use the macros in the xMacroBundle for high-precision computing. Since it is unlikely that you will use them routinely, they are not included in either the toolbar or the menu, but you can of course include them if you wish, or make a new toolbar and/or menu for them. At any rate, you can always access them with Tools > Macro > Macros or Alt + F8.

New x-macros may appear on this site as they become available and have undergone some provisional testing. So far the only macros in this collection are **xLS**, **xWLS**, **xLSPoly**, and **xOrtho**, which are the extended-precision equivalents of LS, WLS, LSPoly, and Ortho respectively, plus **InsertXMBToolbar** and **RemoveXMBToolbar**.

Least squares macros

LS is a general least squares fitting routine for linear, polynomial, and multivariate fitting, assuming one dependent variable. LS0 forces the fit through the origin, LS1 does not. The output provides the parameter values, their standard deviations, the standard deviation of the fit to the function, the covariance matrix and, optionally, the matrix of linear correlation coefficients.

LSPoly applies LS to fitting data to a polynomial of gradually increasing order (up to 14), including criteria (s_f, F-test) useful for deciding how many terms to include in an analysis.

LSMulti applies LS to an increasing number of terms of a multivariate least squares analysis.

LSPermute computes the standard deviation of the fit for all possible permutations of multivariate parameters of up to six terms.

Ortho provides a Gram-Schmidt transformation.

ELS provides least squares smoothing and differentiation for an equidistant (in the independent variable) but otherwise arbitrary function using a ‘Savitzky-Golay' moving polynomial fit. ELSfixed uses a user-selected, fixed-order polynomial, ELSauto self-optimizes the order of the fitting polynomial as it crawls along the function.

WLS is the equivalent of LS with the inclusion of user-assigned weights.

SolverAid provides uncertainty estimates (standard deviations, the covariance matrix, and optionally the matrix of linear correlation coefficients) for Solver-derived parameter values.

Propagation computes the propagation of uncertainty for a single function, for various independent input parameters with known standard deviations, or for mutually dependent parameters with a known covariance.

FOURIER TRANSFORM AND (DE)CONVOLUTION MACROS

FT is a general-purpose Fourier transform macro for forward or inverse Fourier transformation of 2^n data where *n* is an integer larger than 2.

(De)convolve provides general convolution and deconvolution. The convolution macro is generally applicable, the deconvolution macro is not.

(De)ConvolveFT yields convolution and deconvolution based on Fourier transformation.

DeconvolveIt performs iterative (van Cittert) deconvolution. DeconvolveIt0 has no constraints, DeconvolveIt1 assumes that the function is everywhere non-negative.

Semi-integrate & semi-differentiate comprises two small macros for cyclic-voltammetric (de)convolution assuming planar diffusion.

Gabor provides time-frequency analysis.

MISCELLANEOUS MACROS

InsertMBToolbar provides easy access to the macros of the MacroBundle.

RemoveMBToolbar

InsertMBMenu gives an alternative to the MBToolbar, less convenient but also taking up less screen space.

RemoveMBMenu

MovieDemos from my Advanced Excel book.

RootFinder finds a single real root by bisection.

SolverScan lets Solver scan a two-dimensional array of parameter values.

ColumnSolver applies Solver line-by-line to column-organized data.

Mapper generates colored or gray-scale 2-D maps.